Japanese Patent Application Laid-Open No. 7-152920

(54) [Title of the Invention]

GRAPHIC PROCESSOR AND METHOD THEREOF

(57) [Abstract]

[Object]

To solve an acute end and an overlap problem and to enable high-speed software processing.

[Constitution]

An instruction means 1 instructs the kind, size, and modification of a graphic to be processed. Then outline data on the graphic instructed by the instruction means 1 are supplied from an outline data memory 2 to an outline raster scanning means 3. The outline data are so set that the direction of external loop data is opposite to the direction of internal loop data. Then the outline raster scanning means 3 makes a raster scan. For this raster scan, one outline point forming one outline is determined on one line and generated in a painting-out buffer 5. Further, a painting-out means 4 performs an additive subtractive painting-out process and the value of respective pixels of the painting-out buffer 5 are increased or decreased; when the value becomes larger than 1, the pixels are painted out on each line. Lastly, the graphic stored in the paining-out buffer 5 is displayed at a display means 6.

[Claims for the Patent]

[Claim 1]

A graphic processor comprising:

an instruction means for instructing at least the kind and size of a graphic;

an outline data storage means for storing the outline data so set that the direction of external loop data indicating an external outline in which the graphic is a loop inside is opposite to the direction of internal loop data indicating an internal outline in which the graphic is a loop outside;

an outline raster scanning means for making a raster scan for the outline indicated by said outline data by determining one outline point on each line, in which said outline data of the graphic instructed by said instruction means is supplied from said outline data storage means;

a painting-out means for incrementing or decrementing the value of each pixel on each line in one predetermined direction partitioned by said outline, depending on whether said outline determined by said outline raster scanning means is increasing or decreasing in a direction perpendicular to the line processing direction; and

a painting-out buffer for storing the graphic painted out by said painting-out means.

[Claim 2]

A graphic processing method for painting out the inside or outside of a graphic in converting outline data indicating the outline of the graphic into bit map data, comprising:

making a raster scan for the outline indicated by said outline data so set that the direction of external loop data indicating an external outline in which the graphic is a loop inside is opposite to the direction of internal loop data indicating an internal outline in which the graphic is a loop outside by determining one outline point on each line; and

incrementing or decrementing the value of each pixel on each line in one predetermined direction partitioned by said outline, depending on whether said outline is increasing or decreasing in a direction perpendicular to the line processing direction.

[Claim 3]

A graphic processing method for use in the graphic processor according to claim 1, wherein data of one pixel is represented in a signed integer of plural bits or an unsigned integer with an offset, and data of plural pixels is stored in one data word of painting-out means and incremented or decremented in a unit of said one data word or its integral multiple.

[Claim 4]

The graphic processing method according to claim 3, comprising making an increment operation by adding an addend of all clearing a part of plural bits corresponding in a direction opposite to the line processing direction from a part of plural bits corresponding to a initiation point of painting-out, and setting the least significant bit of each of the part of plural bits corresponding to the initiation point of painting-out and a part of plural bits corresponding in a forward direction of the line processing direction from said part of plural bits corresponding to the initiation point of painting-out and clearing each of the other bits, and making a decrement operation by adding an addend of all setting a part of plural bits corresponding in a direction opposite to the line processing direction from a part of plural bits corresponding to the initiation point of painting-out, and if the part of plural bits corresponding to the initiation point of painting-out is not at the least significant level of a data word, clearing the least significant bit and setting the other bits of the part of plural bits corresponding to the initiation point of painting-out, and if the part of plural bits corresponding in a forward direction of the line processing direction from said part of plural bits corresponding to the initiation point of painting-out is not at the least significant level of the data word, clearing the least significant bit and setting the other bits, in which a part of plural bit at the least significant level are all set.

[Detailed Description of the Invention]

[0001]

[Industrial Application Field]

The present invention relates to a graphic processor and a method thereof, and more particularly to a graphic processor and a method thereof for painting out the outline and the inside and outside of a graphic in converting outline vector data indicating the outline of the graphic such as a character into dot data.

[0002]

[Conventional Art]

Conventionally, a method for painting out the inside or outside from the outline of a character drawn by holding the outline of the character as outline data and converting the outline data into dot data is employed to display the character at high quality in scale up or down.

[0003]

And as a method for displaying a coded character in up or down scale on an output device such as a CRT, Japanese Patent Publication No. 53-15624 "Method for generating the character" and Japanese Patent Publication No. 53-41017 "Method for generating the high quality character" are well known. However, these methods have a problem that when a part of acute end in the outline of character is one dot, the painting-out of character is not correctly performed. In the following patents, several methods for solving this acute end problem have been proposed.

[0004]

As a method for painting out the outline vector data of character or graphic by solving the acute end problem, "Image processor" in Japanese Patent Application Laid-Open No. 2-59872, "Graphic painting-out method" in Japanese Patent Application Laid-Open No. 4-684, and "Outline data drawing device" in Japanese Patent Application Laid-Open No. 3-233689 have been proposed. And the "Image processor" in Japanese Patent Application Laid-Open No. 2-59872 is a relatively excellent method that solves the acute end problem in a simple manner suitable for the software processing. This method comprises sequentially performing the reversal painting-out in a raster scan direction while plotting the outline, whereby the painting-out is performed with a simple configuration by avoiding the acute end problem. However, since the reversal process is required locally on one scan line many times, there is a problem that the processing speed is decreased. Thus, in this proposal, a painting-out processing area is divided into two near the center line of graphic, and the painting-out direction in each area is changed, thereby reducing the extent that the reversal process overlaps locally. Even in this case, however, there was still the problem that the processing speed is decreased.

[0005]

Another painting-out problem is an outline overlap problem. This overlap problem will be described below, taking the thickness variation of character that is one modification of outline character as an example. For example, when katakana "□" is defined by an outline vector as shown in Figure 10A, one exemplary method for making the thickness variation involves moving the vertex coordinate in a direction of dividing an angle made by two sides contact at each vertex in two, as shown in Figure 10B. At this time, a collision or overlap of the outline may occur, like a part indicated by the bold arrow in Figure 10B. In this case, the method of Japanese Patent Application Laid-Open No. 2-59872 has the following problem.

[0006]

That is, referring to Figures 11A to 11E showing the views of enlarging the part where this overlap of the outline occurs, a reversal painting-out procedure for painting out the inside of character surrounded by the outline will be described below. First of all, the right side of the leftmost outline is reversed in black, as shown in Figure 11A. Then, the right side of the adjacent outline is reversed in white, so that only the part surrounded by the outlines is made black, as shown in Figure 11B. However, if the same process is performed, an overlap portion of the part surround by the outlines is reversed again, as shown in Figures 11C and 11D, so that a white void part arises though it is surrounded by the outlines, as shown in Figure 11E.

[0007]

In the "Graphic painting-out method" in Japanese Patent Application Laid-Open No. 4-684, a pixel (picture element) is represented in combination bits of an outline flag and a direction bit, whereby the overlap problem caused by scale up or down is solved by rewriting while referring to the pixel during the outline drawing and during the painting-out (the outline drawing and the painting-out are individually performed with this method). Particularly, it is possible to cope with missing strokes in the character due to degeneration of parallel outlines caused by the scale down of character, as shown in Figure 12.

[0008]

However, if the outline drawing is performed by this method, there is necessarily a case leaking out of an update regulation on the directional information of pixel after the start of drawing the outline. For example, in this update regulation, because the direction value for the current pixel is determined by the data of the previous and next pixels, a process immediately after the starting point of the outline is not determined (Figure 13A). Also, if the points without change in the Y coordinate value continue immediately after the starting point of the outline, the indefinite points may consecutively occur (Figure 13B). Accordingly, a process for redetermination by searching for the indefinite point after one cycle of drawing the outline was required, in addition to the procedure as disclosed in Japanese Patent Application Laid-Open No. 4-684.

[0009]

Further, the process from creating the most important directional information to painting-out is not explicitly described in this proposal. However, in the case where a graphic having internally an unpainted portion as shown in Figure 14A has vector data as shown in Figure 14B, if the outline is drawn by determining the direction under the update regulation and redetermining the indefinite point, the result as shown in Figure 16 is obtained. The black circle is the point where the outline bit is set, and the point where the numerical value is not written is the point at the initial state of value (= 0).

[0010]

And if the values for pixels are added in the horizontal direction, the result as shown in Figure 17 is obtained. On the contrary, the conditional judgement of <1> zero, <2> non-zero, <3> even number and <4> odd number is made, and a logical sum with the outline information is taken, whereby the results as shown in Figures 15B to 15E can be obtained for the graphic as shown in Figure 15A. At this time, to correctly paint out the graphic having internally an unpainted portion as shown in Figure 14A, the condition of <4> odd number where the result as shown in Figure 15E is obtained is employed.

[0011]

Herein, a problem where an overlap of the outline as described in the example of thickness variation occurs will be described below. Supposing that a graphic as shown in Figure 18A has vector data as shown in Figure 18B (a view typically showing a situation caused by thickness variation), if the outline is drawn by determining the direction under the update regulation and redetermining the indefinite point in the same manner as in the previous example, the result as shown in Figure 19 is obtained.

[0012]

And if the values for pixels are added in the horizontal direction, the result as shown in Figure 20 is obtained. Herein, the condition of <4> odd number is employed in the same manner as above, the value of the overlap portion becomes -2, which is not the odd number, whereby there is a void in the overlap portion. On the other hand, if the condition of <2> non-zero is employed, the inside of internal loop which is intrinsically not required to paint out is painted out, whereby an essential problem occurs. Further, since this proposal has many judgement processes based on the special rule, and many branches to the special process, including the problem for the indefinite point process that is pointed out, it takes a very long processing time, when the software processing is performed using the device with the constitution as shown in the example of this proposal.

[0013]

And the "Outline data drawing device" in Japanese Patent Application Laid-Open No. 3-233689 uses a method for drawing the outline and painting out individually based on the special rule similar to Japanese Patent Application Laid-Open No. 2-59872 as described above, in which a measure for treating the void problem in the overlap portion of the outline, which is the problem in the above proposal, is described in detail, but the used rule is more complex by introducing the state transition. Therefore, though the higher speed of hardware is one of the purposes, it takes more time to perform the software processing with an MPU (microprocessing unit) in current use, such as a state transition process.

[0014]

[Problems to be Solved by the Invention]

With the method for performing the outline drawing and painting-out separately, the outline is not simply a set of dots, but it is required that the properties for individual parts of the outline are stored in any form to treat the acute end. On the other hand, with the method of Japanese Patent Application Laid-Open No. 2-59872, the acute end problem can be avoided, but the overlap problem can not be avoided.

[0015]

Further, with the conventional method, if both the acute end problem and the overlap problem are avoided, it takes a lot of time to perform the software processing, whereby the method was not very practical. Thus, it is an object of this invention to solve the problem caused by the acute end and outline overlap by a simple method, and to provide a graphic processor and a method thereof that can perform the high speed processing when the software processing is performed by the MPU.

[0016]

[Means for Solving the Problems]

In order to accomplish the above object, there is provided a graphic processor comprising an instruction means for instructing at least the kind and size of a graphic, an outline data storage means for storing the outline data so set that the direction of external loop data indicating an external outline in which the graphic is a loop inside is opposite to the direction of internal loop data indicating an internal outline in which the graphic is a loop outside, an outline raster scanning means for making a raster scan for the outline indicated by the outline data by determining one outline point on each line, in which the outline data of the graphic instructed by the instruction means is supplied from the outline data storage means, painting-out means for incrementing or decrementing the value of each pixel on each line in one predetermined direction partitioned by the outline, depending on whether the outline determined by the outline raster scanning means is increasing or decreasing in a direction perpendicular to the line processing direction, and a painting-out buffer for storing the graphic painted out by the painting-out means. Also, there is provided a graphic processing method for painting out the inside or outside of a graphic in converting outline data indicating the outline of the graphic into bit map data, comprising making a raster scan for the outline indicated by the outline data so set that the direction of external loop data indicating an external outline in which the graphic is a loop inside is opposite to the direction of internal loop data indicating an internal outline in which the graphic is a loop outside by determining one outline point on each line, and incrementing or decrementing the value of each pixel on each line in one predetermined direction partitioned by the outline, depending on whether the outline is increasing or decreasing in a direction perpendicular to the line processing direction.

[0017]

[Operation]

The acute end problem is treated by performing the outline drawing and painting-out in parallel without storing the properties of the outline specially. Also, the external loop data and the internal loop data stored in the outline data are clearly distinguished depending on the sense of loop. An additive subtractive process for each outline point on one line is performed, and a void of the overlap portion is prevented by judging whether or not painting out is performed based on the positive or negative value of each pixel. Further, the processing speed is increased by using a simple algorithm without using the special judgement in a state where plural pixels are integrated into the word length easily processed by the MPU.

[0018]

[Embodiments]

One embodiment of a graphic processor and a method thereof according to the present invention will be described below with reference to the drawings. Figure 1 is a block diagram showing one embodiment of the graphic processor according to the invention. In Figure 1, instruction means 1 instructs the kind of a graphic to be processed (which graphic is processed), the up or down scale, and the modification such as thickness or shape. And outline data of the graphic instructed by this instruction means 1 is supplied from an outline data memory (outline data storage means) 2 to an outline raster scanning means 3. The outline data of each graphic stored in the outline data memory 2 is an array of vertex data and line kind data so set that the direction of external loop data indicating an external outline judging that the graphic is a loop inside is opposite to the direction of internal loop data indicating an internal outline judging that the graphic is a loop outside. And the outline raster scanning means 3 makes a raster scan for this outline data to create the outline. This raster scan involves determining the outline points forming the outline connecting each vertex one by one on each line (of a painting-out buffer 5) and drawing them via a painting-out means 4 in the painting-out buffer 5, as will be described later. Further, the painting-out means 4 performs an additive subtractive painting-out process, as will be described later, and the value of each pixel (picture element) of the graphic drawn in the painting-out buffer 5 is incremented or decremented. If the value becomes a predetermined value or less, or a predetermined value or more, the pixel is painted out on each line, thereby painting out the graphic. Lastly, the graphic stored in the painting-out buffer 5 is displayed at a display means 6. The graphic instructed by the instruction means 1 can be displayed.

[0019]

Figure 2 shows a specific example of the graphic processor according to the invention. Describing this specific example in contrast to the block diagram as shown in Figure 1, a keyboard 1a corresponds to the instruction means 1 for instructing the kind, size and modification of the graphic to be processed, and an MPU (microprocessing unit) 7 is a circuit for drawing and painting out the graphic by making the arithmetic operation on outline data supplied from an outline data ROM 2a in accordance with a program stored in a program ROM 8, and outputting data to a painting-out buffer RAM 5a such as VRAM. The outline data ROM 2a corresponds to the outline data memory 2 of Figure 1, and the MPU 7 and the program ROM 8 correspond to the outline raster scanning means 3 and the painting-out means 4. And the painting-out buffer RAM 5a corresponds to the painting-out buffer 5. The graphic stored in the painting-out buffer RAM 5a is converted into an analog video signal by a DA converter 9 and outputted to a display means 6 such as a CRT.

[0020]

The raster scan for the outline of the graphic that is a principal part of the invention and a painting-out method will be specifically described below. In the drawings from Figure 3, a two-dimensional image plane generated within the painting-out buffer 5 is represented in XY coordinates, the increase direction of X coordinate is the direction where the painting-out process is performed, the increase direction of Y coordinate is the upper direction of character or graphic, and the increase direction of X coordinate is the right direction of character or graphic. Also, the outline data stored in the outline data memory 2 and representing the outline of character to be painted out may be general outline data represented in the straight line or approximate curve, but it is required that the array direction of the data is opposite between the external loop and the internal loop to clearly distinguish them. Since the outline data is generally associated with an expansion processing system, there is no problem with such data.

[0021]

(1) First of all, the creation of outline will be described below. To create the outline from the outline data, a so-called DDA (Digital Differential Analyzer) algorithm is used. This DDA algorithm has been described in various publicly known literatures within Japan, and introduced in detail in "Practical computer graphics - basic procedure and application", supervised by Fujio Yamaguchi, for example. The DDA algorithm involves

representing the solution of a differential equation expressing the straight line in the form of a recurrence formula to make a raster scan for the straight line, and determining the outline point while increasing the value of X or Y pixel by pixel. And an integer type Bresenham algorithm in the DDA algorithm is advantageously treated in a general digital processing system because the decimal fraction is not used and no division is made. Also, it is broadly used because the straight line image with less error can be plotted.

[0022]

This Bresenham generalized algorithm is generalized so that the brightness of the straight line drawn on the CRT may be visually constant without regard to the sense of the straight line. And the sense of the straight line is considered in terms of an octant on the XY plane, and four kinds of <1> select X as the variable of algorithm, <2> select Y, <3> increment the selected variable, and <4> decrement are associated with each pair of second and third, fourth and fifth, sixth and seventh, and eighth and first octants.

[0023]

Though in this embodiment the Bresenham algorithm is used, its purpose is not to make the brightness of the drawn straight line visually constant without regard to the sense of the straight line, but to determine the starting point of the painting-out process. And when the process for all the four kinds is considered, the number of initiation point of painting-outs determined at a time is not determined one as seen from the painting-out direction, causing an inconvenience. Therefore, in this invention, only Y is the variable without regard to the direction of straight line vector, and simply incremented (processing contents corresponding to the pair of second and third octants of the integer type Bresenham algorithm), thereby obtaining the starting point.

[0024]

Herein, the relation between the DDA algorithm and the painting-out will be described below. Figure 3A is an example of the processing result corresponding to the first octant of the integer type Bresenham generalized algorithm, and Figure 3B is an example of the processing result corresponding to the second octant. And in Figure 3A, the outline points are determined while incrementing X in the positive direction, whereas in Figure 3B, the outline points are determined while incrementing Y in the positive direction. When the painting-out process is performed in the positive direction of X with the outline points determined in this manner as the starting point of the painting-out process, a plurality of outline points exist for one Y coordinate value (same line) forming the single outline in Figure 3A, whereby a plurality of initiation point of painting-outs exist to cause an inconvenience. As a result, the processing is not only overlapped but also the inversion or additive subtractive result may possibly have an error.

[0025]

On the contrary, only one outline point exists for one Y coordinate value forming the single outline in Figure 3B, whereby one initiation point of painting-out is determined without inconvenience. Thus, in this embodiment, it is not judged which octant the outline data belongs to, but the DDA algorithm is varied in use such that the variable is only Y coordinate associated with the pair of second and third octants. With such association, the brightness is not necessarily constant, but the outline point resulting from the processing is used as the initiation point of painting-out, whereby there is no problem with the change in the brightness of the outline.

[0026]

(2) The painting-out will be described below mainly about the acute end problem. A painting-out method of this embodiment involves drawing the outline by determining one outline point on each line, using a varied DDA algorithm, to make a raster scan, and incrementing (increasing) the value of each pixel on the right side partitioned by the outline on each line when the outline increases in a direction perpendicular to the line processing direction, or decrementing (decreasing) the value of each pixel when the outline decreases in this direction, in which if the value becomes 1 or more, the pixel is painted out.

[0027]

More specifically, first of all, each pixel is represented in the integer value of several bits, in which the initial value is 0. Then, in the varied DDA algorithm, every time one initiation point of painting-out is determined, the additive subtractive operation of each pixel value is performed in the increasing direction of the X coordinate value. And in making the varied DDA algorithm process, if the Y coordinate value of the starting point of the straight line is smaller than the Y coordinate value of the endpoint, the initiation point of painting-out is determined from the starting point toward the endpoint. At this time, an operation for each pixel is increment (increase). If the Y coordinate value of the starting point of the straight line is larger than the Y coordinate value of the endpoint, the initiation point of painting-out is determined from the endpoint toward the staring point. At this time, an operation for each pixel is conversely decrement (decrease).

[0028]

An instance of painting out the graphic including the acute end by this method will be described below. A triangle composed of three points A(0,0), B(4,8) and C(7,0) is considered. This triangle is external loop, and defined clockwise. First of all, plotting the outline points corresponding to the line segment AB and the painting-out process are performed, as shown in Figure 4. The outline point is determined at each point indicated by the black circle in the direction of the arrow in Figure 4, and every time one outline point is determined, the additive subtractive process for the values of the outline point and all the points (white circle points) on the right side of the outline point is performed. In this case, since the line segment is in the increase direction of the Y coordinate, the additive process is performed, whereby the value of each point on the right side is +1.

[0029]

Next, determining the outline points corresponding to the line segment BC and the painting-out process are performed as shown in Figure 5. Though the outline is defined in the sense from point B to point C, the outline is drawn inversely in the direction of the arrow (direction from point C to point B). The outline point is determined at each point of the black triangle, and every time one outline point is determined, the additive subtractive process for the values of the outline point and all the points (white triangle points) on the right side of the outline point is performed. In this case, the line segment is inversed to the increase direction of the Y coordinate, the subtractive process is performed. As a result, the value of each point is +1 at only the black point and 0 at other points as shown in Figure 6. Also, since the Y coordinate values of point C and point A are equal, it is not required to process the line segment CA.

[0030]

If the painting-out is performed in this manner, there occurs no inconvenience of painting out that the acute end has a lasting effect to the right when the graphic including the acute end is painted out, whereby the accurate painting-out can be performed. Herein, though the loop condition of the DDA algorithm for one line segment is Y variable < Y coordinate of the endpoint in this embodiment, the condition may be Y variable ≤ Y coordinate value to obtain the same result.

[0031]

(3) The painting-out will be described below mainly about the overlap problem. A process for graphic overlap occurring when the katakana "□" as defined by an outline vector indicated in Figure 10A for use in the conventional example is thickened as shown in Figure 10B will be described below. The enlarged views of a part where an overlap of the outline as indicated by

the thick arrow of Figure 10B occurs are shown in Figures 7A to 7E. Herein, the same method that combines the varied DDA algorithm and the additive subtractive painting-out as used in (2) is used.

[0032]

First of all, if the line segment A as shown in Figure 7A is processed, the left side of this line segment is kept at the initial value 0 and the right side of the line segment A is +1, because the line segment A is in the increase direction of the Y coordinate value. If the line segment B is processed as shown in Figure 7B, the part between the line segment A and the line segment B is kept at +1, and the right side of the line segment B is balance 0, because the line segment B is in the decrease direction of the Y coordinate value.

[0033]

Further, if the line segment C as shown in Figure 7C is processed, the pixels on the right side of the line segment C are increased by +1, because the line segment C is in the increase direction of the Y coordinate value. As a result, a part sandwiched between the line segment A and the line segment B (part between the line segment C and the line segment B) on the right side of the line segment C is +2, and the other part (part on the right side of the line segment C and the line segment B) is +1. Also, a part sandwiched between the line segment A and the line segment B (part on the left side of the line segment C and the line segment B) on the left side of the line segment C is kept at +1, and the other part (part between the line segment B and the line segment C) is kept at 0.

[0034]

And if the line segment D as shown in Figure 7D is processed, all the pixels on the right side of the line segment D are increased by +1, because the line segment D is in the increase direction of the Y coordinate value, like the line segment C, whereby a part sandwiched between the line segment A and the line segment B on the right side of the line segment D is +2, and the other part is +1. Lastly, if the line segment E is processed as shown in Figure 7E, all the pixels on the right side of the line segment E are decreased by 1 so that the value is balance 0, because the line segment E is in the decrease direction of the Y coordinate value. And if the positive (1 or greater) value is obtained through this process, the pixel is painted out, whereby the correct painting-out result can be obtained even in the part where the outlines overlap.

[0035]

Accordingly, using the method that combines the varied DDA algorithm and the additive subtractive painting-out, both the acute end problem and the overlap problem can be solved. Also, though the external loop data indicating the external outline in which the graphic is the loop inside has been described above in the embodiment, in the case of the internal loop data indicating the internal outline in which the graphic is the loop outside, the correct painting-out can be performed through the exactly same process, if the outline data is held so that the array direction of the data is opposite to the direction of the external loop data.

[0036]

(4) Lastly, a process for making the painting-out faster will be described below. Herein, the method for making the painting-out faster especially in the software processing using the MPU will be described below. In the painting-out method of the invention as described above, it is required to represent one pixel with plural bits to make the additive subtractive operation on each pixel. Actually, how many bits are required depends on the complexity of the character graphic. Usually, in the use for Japanese outline font, about four bits may be sufficient. On the other hand, the processing unit of data treated by the MPU is generally the power of 2, and when the data processing for a relatively large amount of data is performed, the MPU having a data bus width of 16 bits or 32 bits is usually used.

[0037]

Herein, as an example of performing the painting-out process efficiently by combining a plurality of pixels, a process for data represented in 4 bits/pixel using a 16-bit accessible buffer memory is considered. In this case, the pixel arrangement and the memory arrangement can be associated, as shown in Figures 18A and 18B. Figure 18A is a view showing the pixel arrangement when the character image is represented on the buffer memory, and Figure 18B is a view showing the address arrangement of the buffer memory as seen from the MPU. And four pixels are arranged in one word because the pixel of 4 bits is arranged in the memory where one word is 16 bits.

[0038]

Herein, the initiation point (outline point) of painting-out determined by the DDA algorithm is the data word associated with any of four parts of bits 15-12, bits 11-8, bits 7-4 and bits 3-0 on the memory arrangement. On the other hand, the data word corresponding to the right side of the data word associated with the initiation point of painting-out in the pixel arrangement is processed by incrementing or decrementing all the bit parts.

[0039]

Accordingly, painting out of the initiation point is classified into the above four cases, as shown in Figure 9, although the data word corresponding to the right side in the pixel arrangement is processed in the same manner as where the initiation point of painting-out is all associated with the bits 15-12. Providing that the number of pixels for one line in the X-axis direction is an integral multiple of the number (4) of pixels represented by one data word (the case as shown in Figure 9), an association method for the above four cases involves associating them in accordance with the remainder in which the X coordinate value of the determined initiation point of painting-out is divided by the number of pixels represented by one data word, (each line of a, b, c, d, ... an, bn, cn, dn corresponds to the remainder of 0, 1, 2, 3 in Figure 9).

[0040]

In this manner, the association for four cases is performed, and the pixel of the initiation point of painting out and the pixels on the right side are incremented or decremented, whereby if the value finally becomes 1 or more (or 0 or less), the pixel is processed as being painted out.

[0036]

Next, the actual representation method for the value of pixel and the actual additive subtractive method will be described below. Though the value of pixel may be processed in the normal integer representation as described above, the above combination process can be performed efficiently in the software processing by contriving the representation method. This method enables the high speed processing even in the hardware configuration. Usually, the signed integer is represented in binary number, the two's complement representation is mostly used, but may be represented in the unsigned binary-coded integer with offset. Herein, the latter unsigned binary-coded integer representation will be described below (the numerical value for addition or subtraction is in the two's complement notation).

[0041]

When the unsigned binary-coded integer with offset is represented in n bits, the bit code is represented such that the value of 0 is all set except for the most significant bit, the value of $2^0/2$ is all clear, the value of $-((2^0/2)-1)$ is all set. And an example of 4-bit code is as follows.

[0042]

+8	1111
+7	1110
:	:
+1	1000

0	0111
-1	0110
:	:
-7	0000

[0043]

To make the addition or subtraction (increment or decrement) for this value,

The arithmetic operation

increment: +0001

decrement: +1111 (carry is ignored)

can be performed (when the augend value is in two's complement representation, the addend value is the same in the operation as shown below).

[0044]

Referring to Figures 8A and 8B and Figure 9, an instance of performing this arithmetic operation process in combination of plural pixels will be described below. When the combination of four pixels is incremented, the addends are classified into the following four cases according to the position of bit part corresponding to the initiation point of painting-out.

[0045]

Initiation of painting-out addend

	(binary representation 16 bits)
bits 15-12:	0001 0001 0001 0001
bits 11-8:	0000 0001 0001 0001
bits 7-4:	0000 0000 0001 0001
bits 3-0:	0000 0000 0000 0001

[0046]

On the other hand, in the decrement, the carry is propagated to the upper bits, whereby this process is needed to take care of. For example, when the two pixels that are initialized are decremented, the addend is not all set (-1 in two's complement representation), but it is required that the least significant bit is cleared (-2 in two's complement representation), except the pixel part arranged at the least significant level so that

Initial value:	±0	0111		0111
Addend:	-1	1110	carry	1111
Addition result:		1000	←	1000

[0047]

Further, when the position of the bit part corresponding to the initiation point of painting-out is not at the most significant level, it is required that the addend of the bit part (not decremented) corresponding to the left side of the initiation point is all set (-1 in two's complement representation) to offset the carry. For example, when only the lower level pixel of the two pixels that are initialized is decremented,

Initial value:	±0, ±0	0111		0111
Addend:	±0, -1	1111	carry	1111
Addition result:		0111	←	1000

[0048]

Accordingly, when the combination of four pixels is decremented, the addends are classified into the following four cases according to the position of the bit part corresponding to the initiation point of painting-out, whereby the correct decrement is enabled by using the addend corresponding to respective case.

[0049]

Initiation of painting-out addend

	(binary representation 16 bits)
bits 15-12:	1110 1110 1110 1111
bits 11-8:	1111 1110 1110 1111
bits 7-4:	1111 1111 1110 1111
bits 3-0:	1111 1111 1111 1111

[0050]

Though the increment and decrement for the combination of four pixels have been described above, the invention allows the addition subtraction to be performed by acquiring the addend in the same manner, in the case of the other number of pixels and 32 bits, and the increment or decrement can be performed by adding the addend as described above, whereby the high speed arithmetical operation is enabled. Further, since the pixels to be painted out may be determined after acquiring the final values, it is unnecessary to perform the inversion process many times, and the painting-out process can be performed at high speed.

[0051]

[Advantages of the Invention]

The graphic processor and a method thereof of the invention can paint out the acute end correctly without storing the properties of the outline specially, because the outline drawing and the painting-out are performed in parallel.

[0052]

Also, the direction of external loop data and the direction of internal loop data stored in the outline data are opposite and clearly distinguished, the additive subtractive operation for each outline point on one line is performed, and a void of the overlap portion is prevented by judging whether or not painting out is performed based on the positive or negative value of each pixel. Also, the correct painting-out can be performed through the same process without distinguishing the external loop data and the internal loop data.

[0053]

Further, the high speed software processing and the high efficient use of memory are enabled by using a simple algorithm without using the special judgement in a state where plural pixels are integrated into the word length easily processed by the MPU. And the hardware processing can be also implemented in a simple configuration.

[0054]

Also, by treating plural pixels at the same time, there is the effect that it is possible to make effective use of a function of processing the data bit number that is an integral multiple of the data bus width in hardware partly mounted on the MPU in recent years with a single instruction.

[Brief Description of the Drawings]

[Figure 1]

Figure 1 is a block diagram showing one embodiment of a graphic processor according to the present invention.

[Figure 2]

Figure 2 is a block diagram showing a specific example of the graphic processor according to the invention.

[Figure 3]

Figure 3A is a view showing an example of processing result corresponding to a first octant in an integer type Bresenham generalized algorithm and Figure 3B is a view showing an example of processing result corresponding to a second octant in the same algorithm.

[Figure 4]

Figure 4 is a view for explaining the painting-out of a graphic including an acute end according to the invention.

[Figure 5]

Figure 5 is a view for explaining the painting-out of a graphic including an acute end according to the invention.

[Figure 6]

Figure 6 is a view for explaining the painting-out of a graphic including an acute end according to the invention.

[Figure 7]

Figures 7A to 7E are views for explaining an overlap problem of painting-out according to the invention.

[Figure 8]

Figure 8A is a view showing a pixel arrangement when a character image is represented on a buffer memory according to one embodiment of the invention, and Figure 8B is a view showing an address arrangement of the buffer memory as seen from an MPU.

[Figure 9]

Figure 9 is a view showing a pixel arrangement for explaining the classification of painting out by the initiation point according to one embodiment of the invention.

[Figure 10]

Figure 10A is a view showing an example of character, and Figure 10B is a view showing an example in which the character thickness is varied.

[Figure 11]

Figures 11A to 11E are views for explaining the conventional overlap problem of painting-out.

[Figure 12]

Figure 12 is a view for explaining the conventional example.

[Figure 13]

Figures 13A and 13B are views for explaining a problem with the conventional example.

[Figure 14]

Figure 14A is a view showing an example of a graphic, and Figure 14B is a view showing a vector example.

[Figure 15]

Figure 15A is a view showing an example of a graphic, and Figures 15B to 15E are views showing the painting-out result examples of the graphic.

[Figure 16]

Figure 16 is a view showing an example of drawing the outline of the graphic as shown in Figure 14.

[Figure 17]

Figure 17 is a view showing an example of drawing the outline of the graphic as shown in Figure 14.

[Figure 18]

Figure 18A is a view showing an example of a graphic, and Figure 18B is a view showing a vector example.

[Figure 19]

Figure 19 is a view showing an example of drawing the outline of the graphic as shown in Figure 18.

[Figure 20]

Figure 20 is a view showing an example of drawing the outline of the graphic as shown in Figure 18.

[Description of Symbols]

1	instruction means
1a	keyboard
2	outline data memory (outline data storage means)
2a	outline data ROM
3	outline raster scanning means
4	painting-out means
5	painting-out buffer
5a	painting-out buffer RAM
6	display means
7	MPU (microprocessing unit)
8	program ROM
9	DA converter

Figure 1

1	Instruction means
2	Outline data memory
3	Outline raster scanning means
4	Painting-out means
5	Painting-out buffer
6	Display means

Figure 2

1a	Keyboard
2a	Outline data ROM
5a	Painting-out buffer RAM
6	Display means
8	Program ROM
9	DA converter

Figure 8

#1	Pixel arrangement
#2	Memory arrangement
#3	Word address
#4	Bit

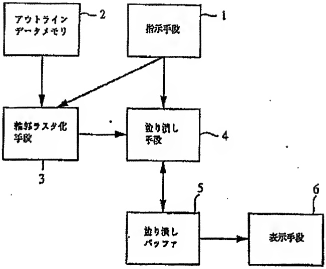
Figure 9

#1	Pixel equivalent to one data word of MPU
#2	Pixel array

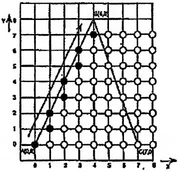
Figure 15

#1	Area

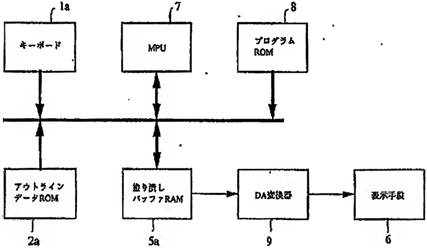
【図1】



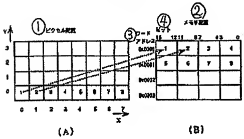
【図4】



【図2】

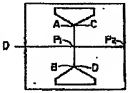


【図8】

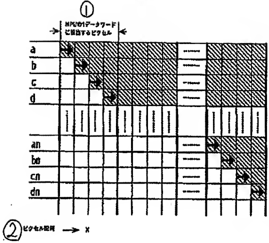


(A)　　(B)

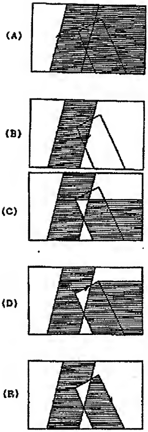
【図12】



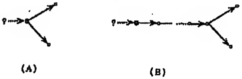
【図９】



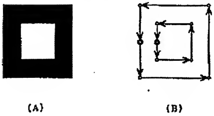
【図１１】



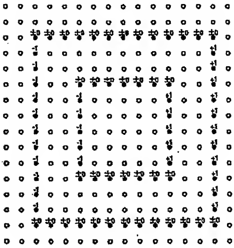
【図１３】



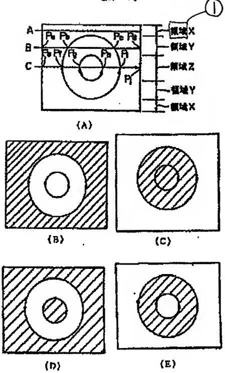
【図１４】



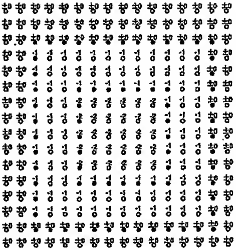
【図１６】



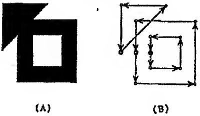
【図15】



【図17】



【図18】



【図19】